NEEDS 298
6/9/03

1387D
BRIEFING
ONLY

(AR 530-1, Operations Security)

I am aware that there is foreign intelligence interest in open source publications. I have sufficient technical expertise in the subject matter of this paper to make a determination that the net benefit of this public release outweighs any potential damage.

Reviewer: MITCH KOZERA 6S-13 Team Leader
Name Grade Title

[signature] 06/05/03
Signature Date

Description of Information Reviewed:

Title: PERVASIVE COMPUTING: WHY DID THE LOGISTICS SOLDIER CROSS THE ROAD?

Author/Originator(s): MARK SALAMANGO

Publication/Presentation/Release Date: 6/8/03

Purpose of Release: CONFERENCE

An abstract, summary, or copy of the information reviewed is available for review.

Reviewer's Determination (circle one):

1. Unclassified Unlimited.

2. Unclassified Limited, Dissemination Restrictions IAW ________.

3. Classified. Cannot be released, and requires classification and control at the level of ________.

Security Office (AMSTA-CS-S):
Concur/Nonconcur [signature] 6-6-03
Signature Date

Public Affairs Office (AMSTA-CS-CT):
Concur/Nonconcur [signature] 6/6/03
Signature Date

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

20060926092

Pervasive Computing: Why did the Logistics Soldier Cross the Road?

Mark J. Salamango
National Automotive Center
U.S. Army Tank-automotive & Armaments Command (TACOM)

ABSTRACT

Enterprise networks are becoming distributed. The central data center has given way to a distributed network environment containing distributed server clusters, edge servers that optimize data process and dissemination, and a new tier of network-enabled devices that provide ubiquitous access. In essence, the network is expanding outward and embracing a series of new processing nodes (e.g., PDAs, cell phones, vehicles, MP3 players, consumer appliances, etc.). These network nodes are called pervasive devices.

The goal of pervasive computing (PvC) is to make data and application services available to any authorized user anywhere, anytime, and on any device. This is accomplished through a robust architecture including software abstraction both at the device level and at the back end.

In part, abstraction is facilitated through the use of the Java Virtual Machine (JVM). A JVM consists of interpreters and a run-time environment which can read java byte code. It enables the construction of machine and operating system (OS) independent applications and services because there are JVMs for most Operating Systems (OSs).

This paper will go over pervasive technology in general, it will review potential considerations, a PvC architecture, and data mining of PvC information. Throughout this paper there are examples of how PvC can be used to impact the Department of Defense (DoD), however its uses are only limited by imagination.

INTRODUCTION

Why did the Logistics soldier cross the road? Obviously he didn't know that with Pervasive Computing, he could get the data he needed from where he was standing.

The mantra of Pervasive Computing from the beginning was to create an environment where information and communication is available everywhere, for everyone, at all times. This environment includes toys, appliances, groceries, vehicles, homes, consumer electronics, and much more.

The focus is on users and their tasks, not necessarily on computing devices and technology. Each device provides a service or tool that when properly configured solves a user's problem, gives content, or feeds further information to another device, set of devices, or some back-end infrastructure.

By "commodifying" the enabling technologies, PvC has cost points well below what they would be if each architecture sub-component was designed from scratch.

For the paradigm of PvC to work, new devices need to be added seamlessly through discovery protocols and the architecture must be able to scale to support millions of PvC nodes.

POTENTIAL PITFALLS

Pervasive devices are inherently heterogeneous. For example, pervasive devices can be appliances, vehicles, PDAs, and much more. Each one of them can be different. Moreover, each device has specific storage, processing, power requirements, interfaces, networking capabilities, and form factors. Consequently applications had to be rewritten to the varying capabilities of each device.

Now with Java Virtual Machines (JVMs) on the device, and strong middleware at both the device level and on the back end, this potential obstacle can be avoided. Instead, we have an architecture that is both a foundation to build affordable applications and an integration platform.

A second potential problem is network connectivity. As more and more 802.11 (Wi-Fi) connections come online, this problem will be less prevalent. However, to further alleviate outages, mobile devices should be able to pick up where the "normal" connection has left off.

Take for instance the notion of an edge server. An edge server is a replicated server that typically resides closer to the user. Because of its proximity, there aren't as many TCP/IP hops needed to get the data to the user. Usually edge servers break up the user base into more manageable segments. In this same way, a vehicle, or device could be considered a Wi-Fi router and gateway, enabling communication to devices around it. Thus fewer outages would be likely because you could dynamically chain together wireless devices to enhance broadband communication ranges.

Keeping track of the multitude of devices is a key concern. Will the infrastructure these devices are built upon scale? There have been enormous strides in the middleware layer (e.g., WebSphere) on both the client and on the back end. The middleware provides core services such as voice, video, and data, as well as security, collaboration, integration, and workflow services. Because of these core services and the development of discovery protocols, powerful servers, and commercial tools, tracking mobile nodes is now possible.

PORTAL TO THE WORLD

Pervasive computing makes information available to users in the way they want to see it. A Logistics Assistance Representative (LAR) may have different needs than a General or even a Sergeant. The LAR may want to know specific information about problems with a vehicle, while the General may just need to know that a vehicle is disabled or how much fighting power is available in theater.

Take for instance the example of a mechanic at an automotive dealership, or a maintenance soldier in the field. The mechanic may use a pervasive device such as a ruggedized PDA, a tablet PC, or another processing device. He could perform diagnostics on the vehicle, perhaps wirelessly, and then assess the fault codes pulled off the vehicle.

E-SUPPLY

Once the diagnosis has been made to distinguish the faulty part, the mechanic can pull up his interface on his pervasive device to immediately order the part. The part could come from either a supply area within the company, or from a 3rd party. The mechanic probably doesn't care, he just needs the part. At the same time, he can prep the vehicle for the repair.

After the part order has been approved through the proper chain of command, back-end servers can make arrangements through intelligent algorithms to find the nearest part, or if necessary, the part can be made in a Rapid Manufacturing Cell. The part can then be shipped to the mechanic or to a depot where repairs can take place.

Once the part has been delivered, the mechanic may need training, or instruction on repairing the vehicle.

E-LEARNING

Ideally, the pervasive device could be used as an e-learning tool. The device should be able to connect to back-end databases which track training levels of each mechanic. Using that information, e-learning tools can guide the mechanic through the repair by giving information in a way that the mechanic has been trained to understand. If the mechanic has advanced training, his interface or portal would reflect his skill level. A step-by-step approach may be unnecessary for some mechanics. In addition, there may be a more efficient, less costly way to fix the part or assembly. If the mechanic has the proper training, he may be lead to attack the problem differently than someone who is new to the job.

For a relatively inexperienced mechanic, he may have to have each step shown to him with audio, rotating pictures, and textual instructions. This is just another example of tailoring a portal to fit the end-user's needs.

Pervasive computing makes information available in different ways. It can be presented on devices with small screens, large screens, and even on devices without interfaces. When interfaces aren't available the device can collect data, manipulate it, and then leave it until someone or something needs it.

ARCHITECTURE

The Device Level

There are five main components necessary at the device level to facilitate the creation of secure and scalable pervasive devices. These are Application Components, User Interface Components, Communication Components, Application Development Tools, and Security Components.

Application Components consist of the Java Virtual Machine (JVM) and Device libraries, Service Management Framework, Embedded HTTP server and associated Enabled Applications, Signature Applications (Direct to GUI, AWT, TTS or Speech Recognition), and Data management (DBMS, Reliable Messaging Synchronization).

While not inherently required for a device to be pervasive, a JVM can add portability to an application that may not be available otherwise. The role of the JVM is to provide interpreters and a run-time environment from which to run java code. This allows for portability of code because there are JVMs for most OSs.

User Interface Components include Graphical User Interface (GUI) widgets, text to speech and speech

recognition components, a shell environment, browsers, and plug-ins for multimedia.

Communication Components deal with the communication and routing of information from device to device, and from the device to the back end. Some components are TCP/IP, WAP, Wi-Fi (802.11), Bluetooth, and other discovery protocols as well.

Of equal importance is the ability to create applications through a robust set of Application Development Tools. These tools allow the user to build applications in the language that is native to the operating system found on the device. If there is a JVM on the device, standard java development tools can be used.

Finally, security of the system is necessary. The Security Components section of the architecture could be comprised of SmartCards, Cryptography, Secure Sockets Layer (SSL), Specific Intercomponent and Device to Device security models, Memory Management Unit (MMU) protected processes, Code signing, Firewalls, and password authentication.

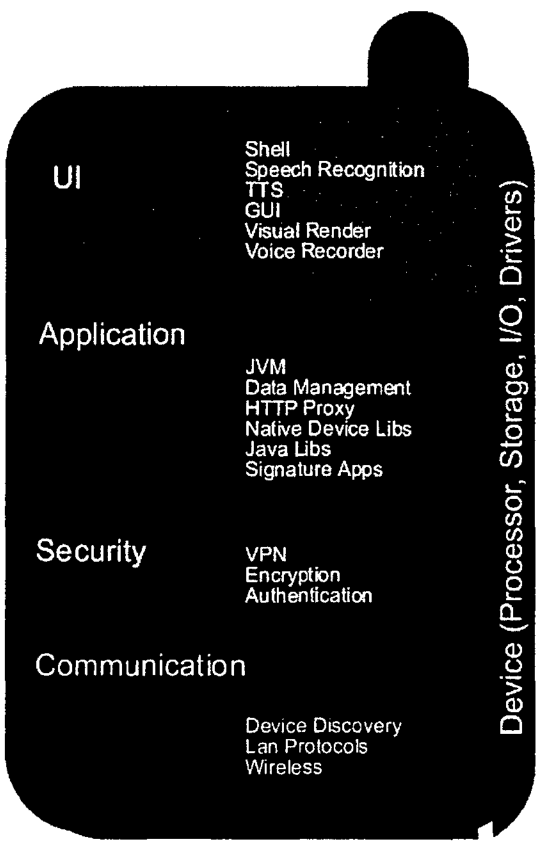


Figure 1: Architecture Components

Much work has gone into tying some of these components into a core Java runtime environment. The result of this work is the MIDP (Mobile Information Device Profile) specification and APIs (Application Programming Interfaces). The specification in concert with the CLDC (Connected Limited Device Configuration) gives core functionality required for mobile applications. Some key functionality given by the MIDP APIs is a user interface, network connectivity, local storage, and management of the application throughout its lifecycle.

Applications that are built on MIDP are called MIDlets. In Figure 2 below, you can see the basic MIDP Architecture.

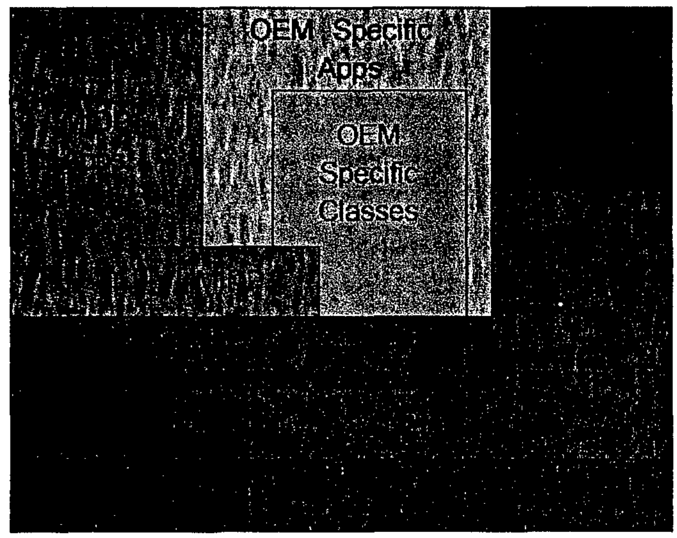


Figure 2: Basic MIDP Architecture

In the Back End

Often neglected in common thinking, the back end architecture is responsible for delivering content, configuration management, authentication, and scalability of the system.

To accomplish these tasks, the back end houses robust databases, web servers, application servers, data servers, network health monitors, and maintains high-availability and disaster recovery mechanisms.

WEB-ENABLED

So far we have looked at the architecture of coding directly to the pervasive device. Another common approach is to use Internet/Intranet-ready devices to deliver applications as dynamically generated web pages.

In this way, pervasive devices can take advantage of commercial applications like Internet Explorer as a front-end delivery mechanism. Therefore, we can use well-defined tools and standards to create content and interactivity to a back-end infrastructure.

Just like most dynamic applications these days, there is a place on the back-end for databases, application servers, web servers, etc.

MINING AND ANALYSIS

The scope of pervasive computing is basically limitless. Anyplace a processor, sensor, or JVM can reside, adds

to the infrastructure of pervasive computing. In the home, processes running on a refrigerator could automatically add items to your grocery list by sending information to a PDA or vehicle of the "shopper" in the family. To take this idea further, when items get low they could be ordered and shipped directly to the consumer.

Once data is captured, analysis and data mining can take place. In the example of a home, diet plans, food consumption, and expenditures can be scrutinized and used for improved health both physically and monetarily.

Another example could be in the case of a vision system which recognizes people in a room. It may discern how many people are in the room and the current outside temperature. Based on the system's data gathering, it could potentially plan heating and cooling of the room.

CONCLUSION

In summary, it should now be apparent that PvC technology has both utility and maturity to be of use to the military. The commercial nature of PvC brings about the commodification of the components necessary to build a complete system. This drives cost down while improving the robustness of the architecture and tool sets.

To further the utility of PvC, back-end services can be taken advantage of on an "on-demand" basis. In essence you pay for what you use.

In a world where eventually everything will be linked, it will be interesting to see who is left crossing the road...it surely won't be the chicken with the PDA.

REFERENCES

http://www.ibm.com

http://www.sun.com

http://www.wirelesssolutionsjournal.com/articles/issue2/IBM-PvC-Story.pdf

CONTACT

Mark Salamango (salamanm@tacom.army.mil) is the Chief Pervasive Architect working for the U.S. Army Tank-automotive & Armaments Command's National Automotive Center in Warren, MI. Currently working in the area of PvC he has obtained a BS in Computer Science from University of Michigan in Ann Arbor. Mark has held various prestigious positions in industry such as IT/IS Director, Level II Technical Consultant, and is the President of his own consulting corporation.

DEFINITIONS, ACRONYMS, ABBREVIATIONS

I have compiled some common acronyms to use as a reference.

DoD: Department of Defense

JVM: Java Virtual Machine

PvC: Pervasive Computing

API: Application Programming Interface

MIDP: Mobile Information Device Profile

CLDC: Connected Limited Device Configuration

TARDEC

UNITED STATES ARMY
NAC
NATIONAL AUTOMOTIVE CENTER

Pervasive Computing: Why did the logistics soldier cross the road?

Mark J. Salamango
Chief Pervasive Architect
USA TACOM
Tel: 586-574-7126
Fax: 586-574-6167
salamanm@tacom.army.mil

TARDEC

Agenda

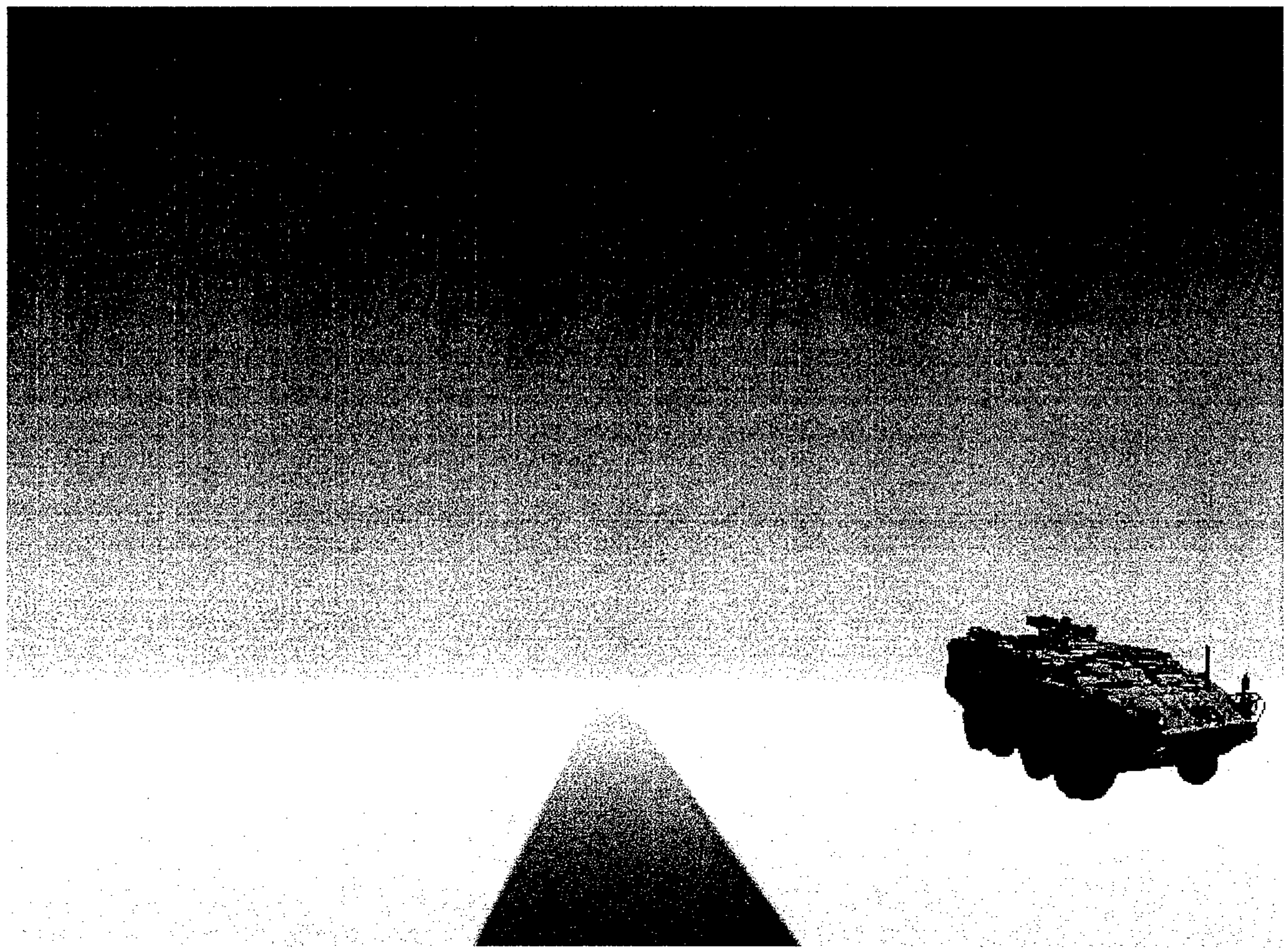
UNITED STATES ARMY
NATIONAL AUTOMOTIVE CENTER

- Introduction: Who is Mark Salamango?
- Why did the logistics soldier cross the road?
- What is Pervasive Computing (PvC)?
- Device Architecture
- The Back End
- Demonstration?
- Conclusion

TARDEC

Why did the logistics soldier cross the road?

UNITED STATES ARMY
NAC
NATIONAL AUTOMOTIVE CENTER



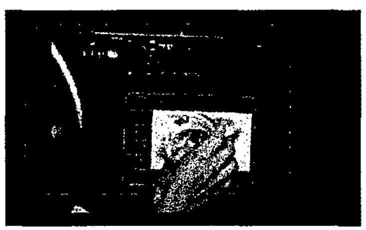
3

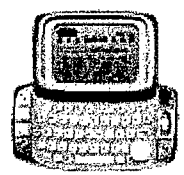
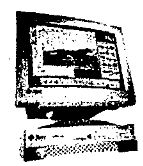
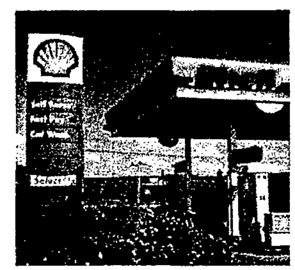
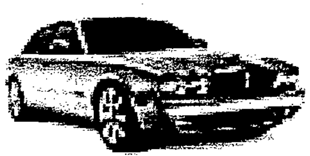
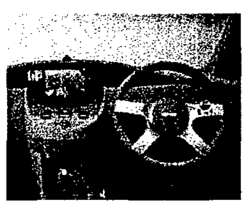
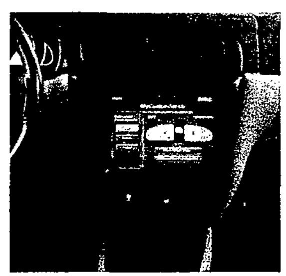
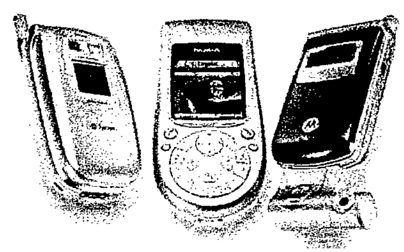
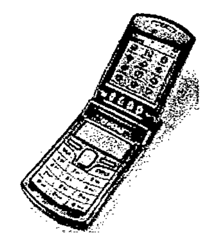
TARDEC

What is Pervasive Computing (PvC)?

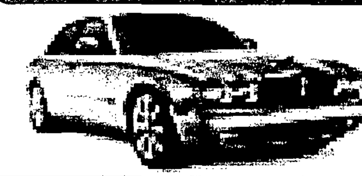
UNITED STATES ARMY NAC NATIONAL AUTOMOTIVE CENTER

- The goal of pervasive computing (PvC) is to make data and application services available to any authorized user anywhere, anytime, and on any device.
- Create an environment where “everything” is a compute node which communicates wirelessly and interacts seamlessly with humans.



Pervasive Computing



5

Pervasive Computing

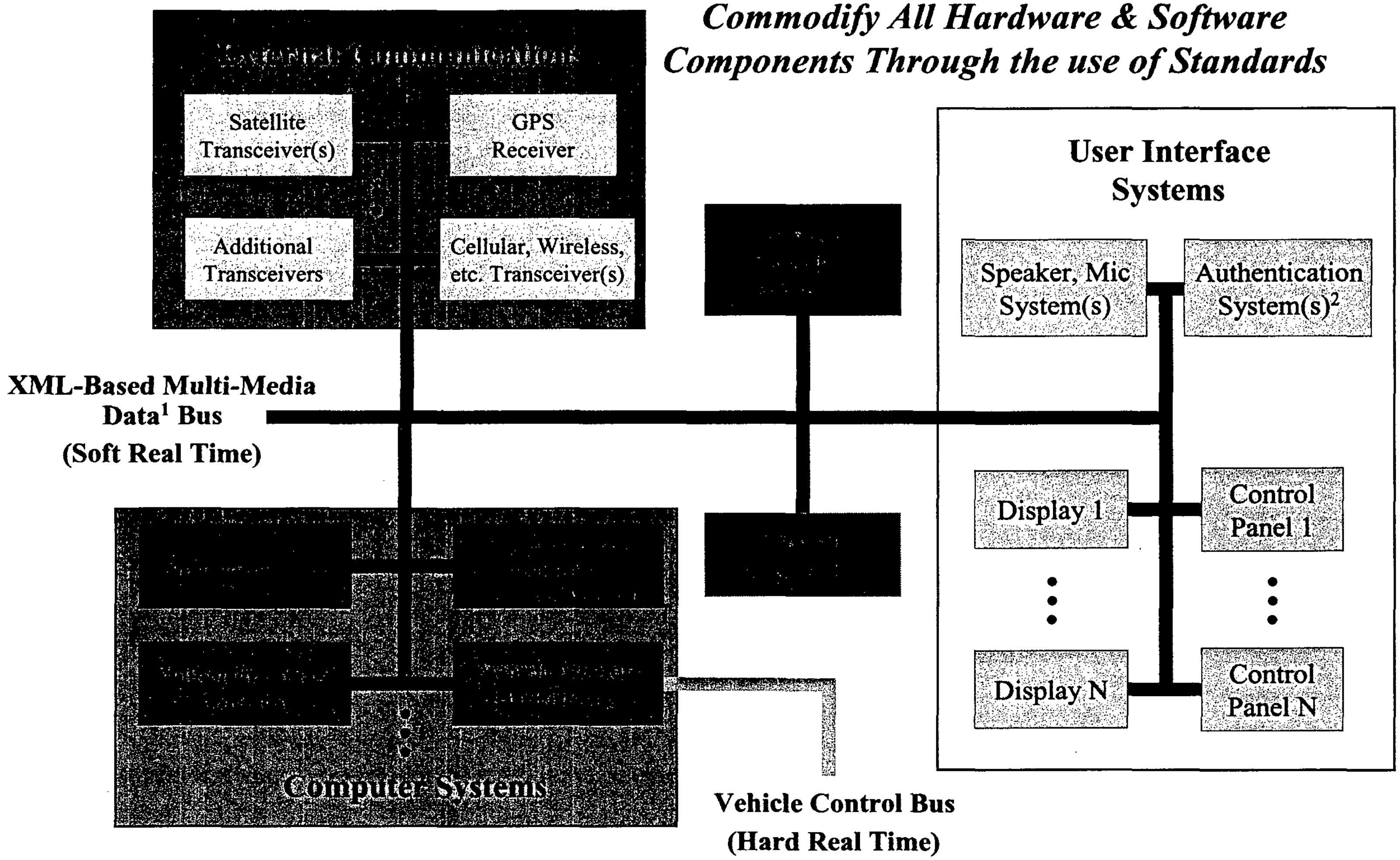
Device Platform

Interim Dual In-Vehicle Architecture

TARDEC

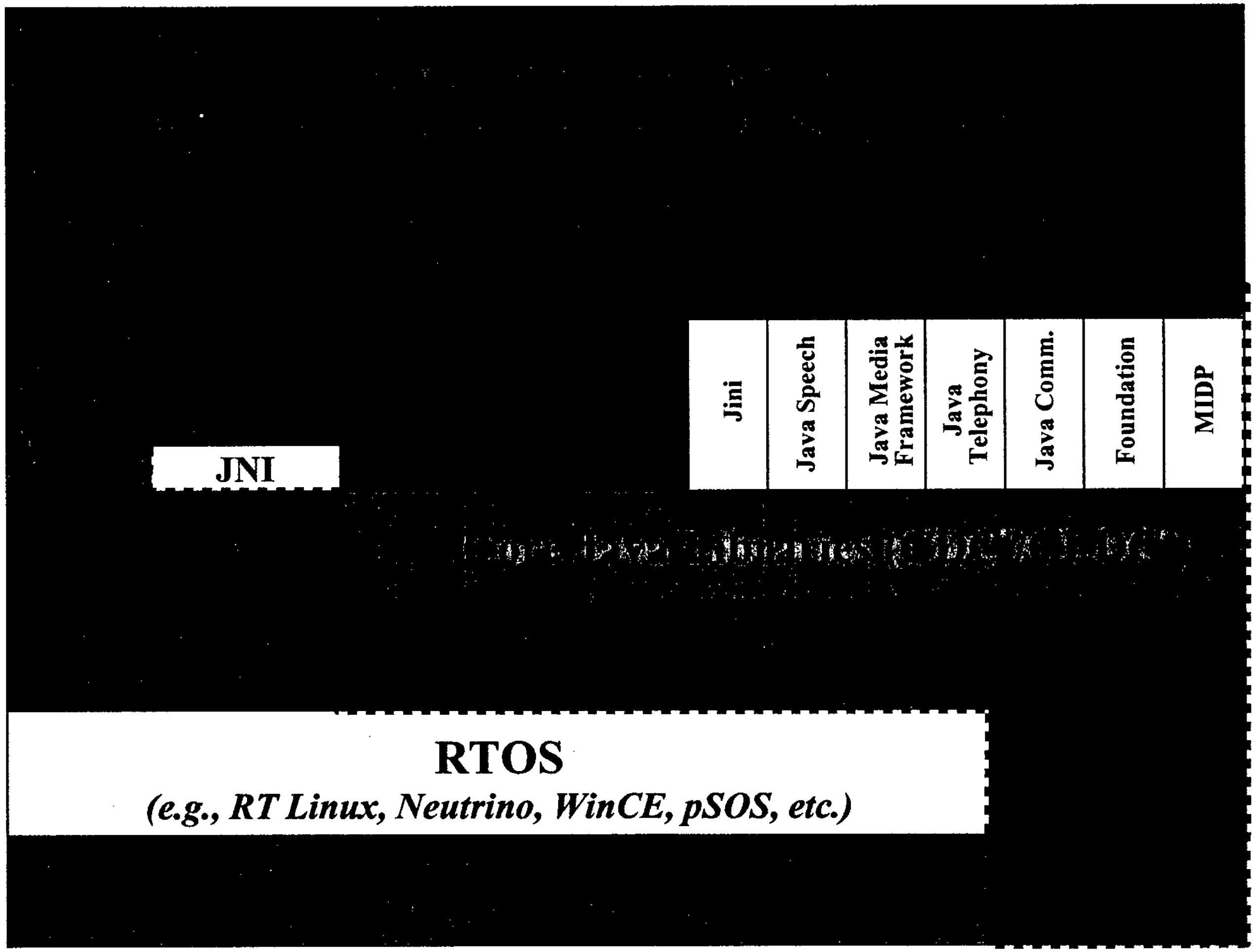
UNITED STATES ARMY NAC NATIONAL AUTOMOTIVE CENTER

Commodify All Hardware & Software Components Through the use of Standards



[1]: Data (Audio, Video, Graphics, Animation, etc.)
[2]: E.g., PDAs, Cell Phones, Games, Java Card, Java Ring, etc., (both active and passive systems)

In-Vehicle
Software Architecture



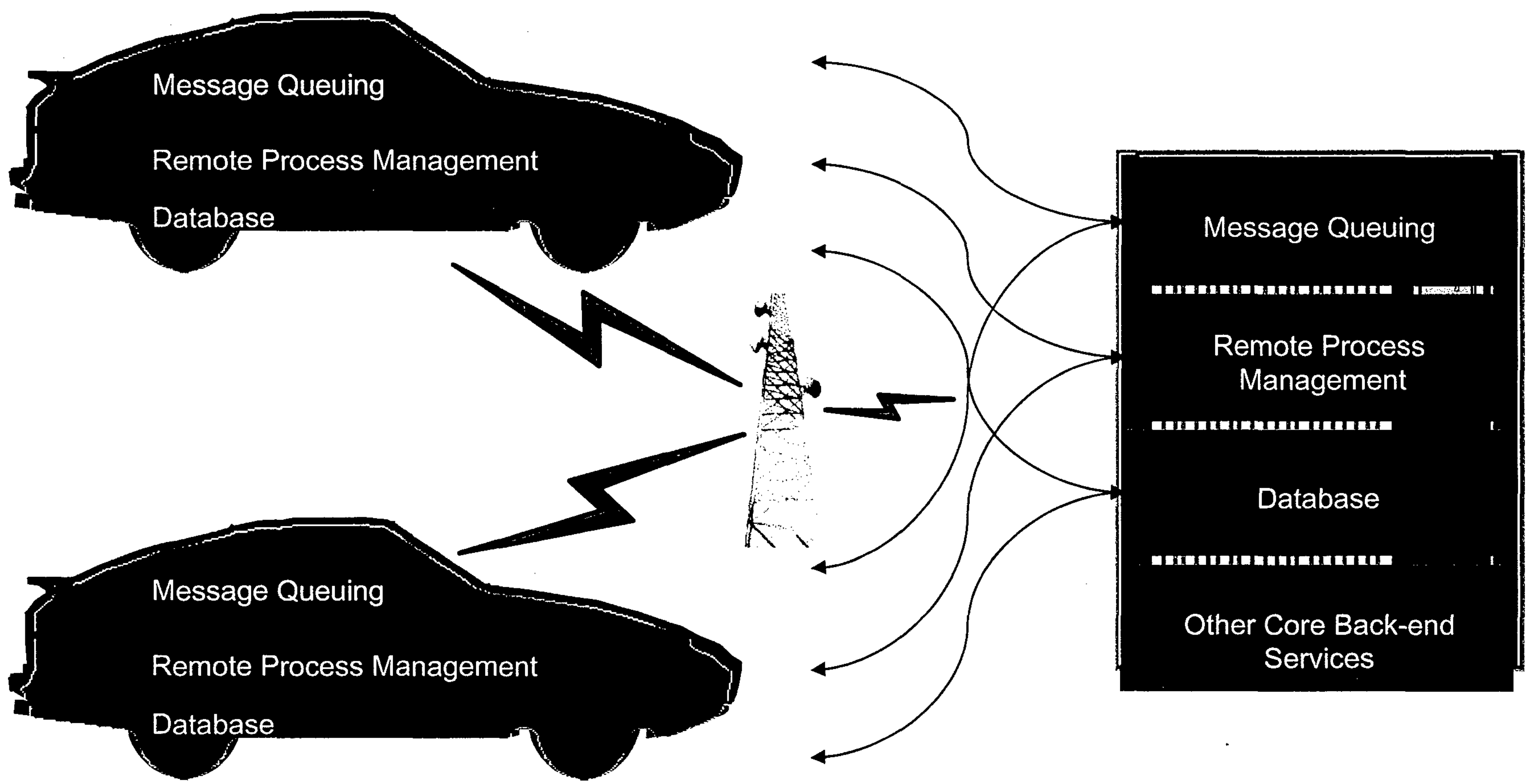
[1]: CDC: Connected Device Configuration
[2]: CLDC: Connected Limited Device Configuration

8

TARDEC

Some Client Server Applications and their Communication

UNITED STATES ARMY NAC NATIONAL AUTOMOTIVE CENTER



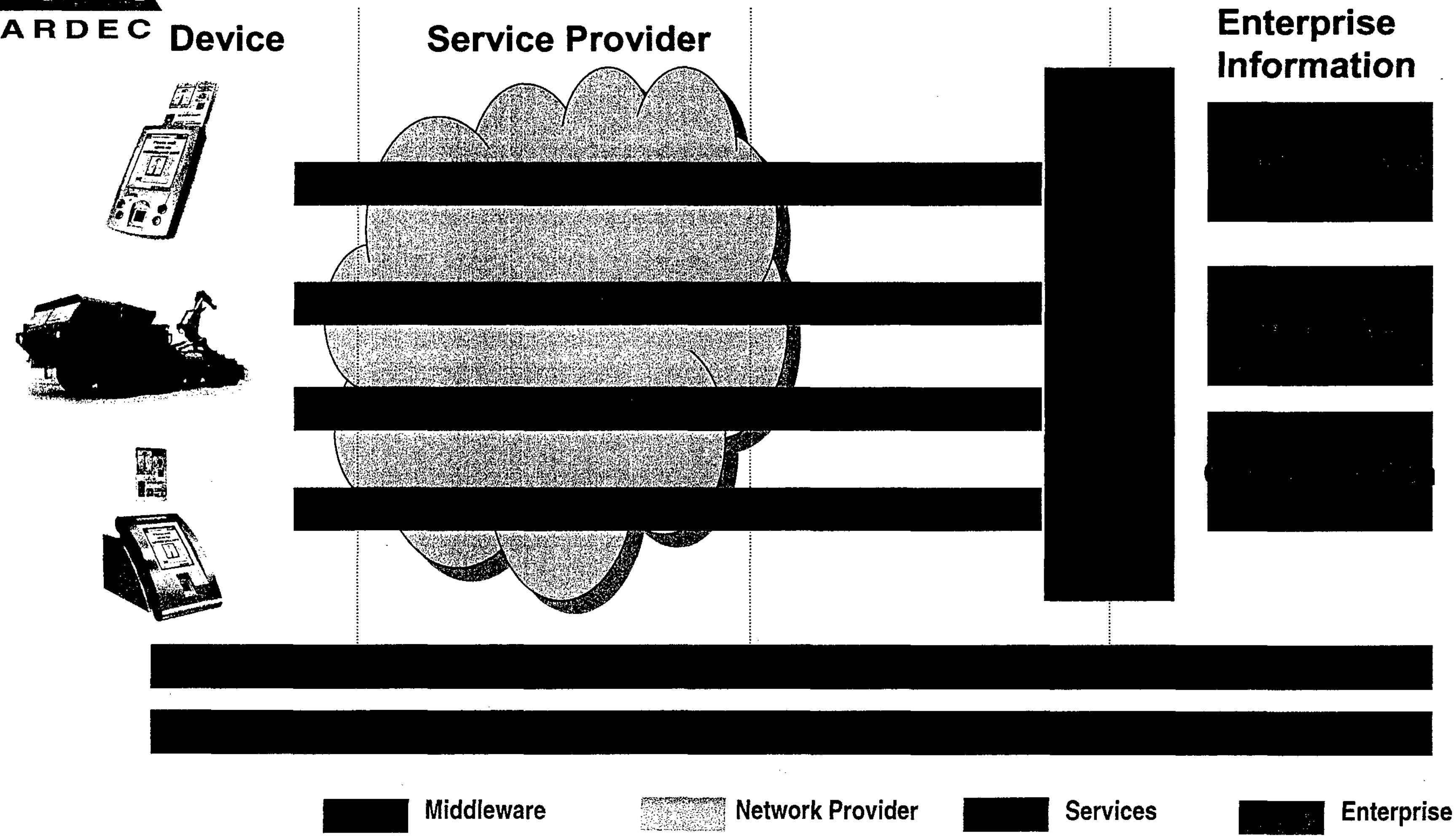
There are many other applications on both the client and server such as diagnostics/prognostics, and many more will evolve!

5

TARDEC

The Total Wireless Solution

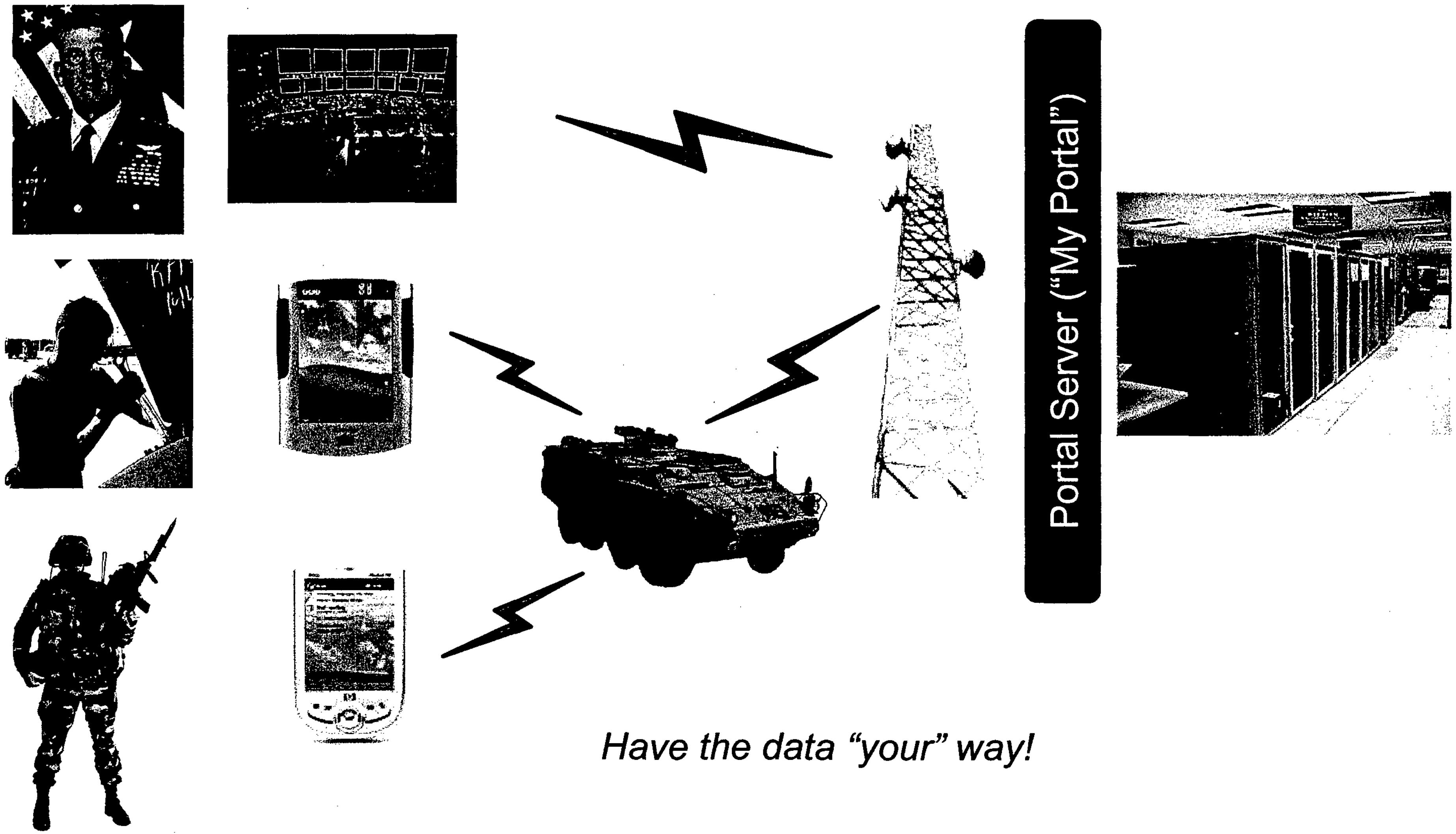
UNITED STATES ARMY
NAC
NATIONAL AUTOMOTIVE CENTER



TARDEC

Man/Machine Interface

UNITED STATES ARMY
NAC
NATIONAL AUTOMOTIVE CENTER



Have the data "your" way!

17

TARDEC

Conclusion

UNITED STATES ARMY NATIONAL AUTOMOTIVE CENTER

- PvC is important!
- It is here, and moving quickly.
- Let's pick an architecture that scales, offers security, and gives us a platform on which to build applications more rapidly.
- We need strong middleware on the device and on the back-end.